

IFML to React Interpreter

USER MANUAL v0.1.0

INFORMASI UMUM

Bagian ini menjelaskan informasi dan tujuan dari sistem yang telah dibuat.

System Overview

IFML to React Interpreter adalah program untuk mengubah model aplikasi web yang telah disusun dalam IFML menjadi aplikasi web berbasis React. Aplikasi web dijalankan pada suatu *server* untuk dapat diakses oleh pengguna. Pengguna dapat mengaksesnya dengan menggunakan *browser* pada komputer.

Ringkasan Manual

User Manual terdiri dari empat bagian: Informasi Umum, Informasi Sistem, Persiapan, dan Penggunaan Program.

- Bagian **Informasi Umum** menjelaskan *hardware* dan *software* pendukung yang diperlukan untuk menjalankan program, akses yang tersedia untuk pengguna.
- Bagian **Informasi Sistem** menjelaskan *hardware* dan *software*

pendukung yang diperlukan untuk menjalankan program, akses yang tersedia untuk pengguna.

- Bagian **Persiapan** menunjukkan tahapan bagaimana program diperoleh dan dipersiapkan untuk dapat digunakan. Tahapan tersebut mencakup instalasi program dan instalasi *software* pendukung pada *environment*.
- Bagian **Penggunaan Program** menunjukkan cara menjalankan *interpreter* dan hasil berupa aplikasi web.

INFORMASI SISTEM

Bagian ini menjelaskan *hardware* dan *software* pendukung yang diperlukan untuk menjalankan program, akses yang tersedia untuk pengguna.

Konfigurasi Sistem

IFML to React Interpreter berjalan pada *platform* yang telah terpasang **Python** versi **3.6.4** atau lebih baru. *Platform* tidak terbatas pada sistem operasi berbasis Unix selama **Python** tersedia pada sistem operasi tersebut.

Program ini tidak memerlukan internet untuk menjalankan tugas.

Internet dibutuhkan ketika sistem operasi tidak tersedia **Python** dengan versi minimal yang dibutuhkan. Untuk mendapatkan **Python** yang

sesuai, pengguna dapat mengakses [halaman unduh Python](#) dan memilih *binary Python* yang sesuai dengan sistem operasi yang digunakan.

PERSIAPAN

Bagian ini menunjukkan tahapan bagaimana program diperoleh dan dipersiapkan untuk dapat digunakan. Tahapan tersebut mencakup instalasi program dan instalasi *software* pendukung pada *environment*.

Instalasi

Berkas program dapat diperoleh pada *repository* utama [affandhia/ifml-pwa](#). Terdapat dua metode untuk memperoleh berkas program.

Metode 1: Mengunduh Berkas **.zip**

Berkas dapat diperoleh dengan mengakses [tautan ini](#). Berkas berupa *file* berekstensi **.zip** yang harus diekstrak. Pada berkas tersebut terdapat satu *folder* bernama **ifml-pwa-release** yang mengandung program.

Bila tautan tidak dapat diakses, berkas dapat diperoleh dengan membuka

halaman utama repository utama

[affandhia/ifml-pwa](#). Pada

halaman utama tekan tombol “Clone or download” lalu tekan tombol

“Download ZIP”.

Metode 2: *Clone Repository* Menggunakan Git

Berkas dapat juga diperoleh dengan menggunakan program bernama **Git** versi **2.20.1** atau lebih. Proses tersebut dinamakan *clone*. Untuk melakukan *clone*, sistem harus menjalankan perintah:

```
git clone https://github.com/affandhia/ifml-pwa
```

Folder **ifml-pwa** akan tersedia ketika **Git** telah selesai melakukan *clone* terhadap *repository*.

Konten dari Folder Program

Program terdiri dari beberapa *folder* dan *file*. Secara umum, keseluruhan berkas akan terlihat seperti susunan dibawah ini:

```
-folder utama-
├─ MANIFEST.in
├─ abs_bankaccount.core
├─ abs_bankaccount.uml
├─ custom_xmi_parser
│   └─ ...
├─ ifml_parser
│   └─ ...
├─ ifml_to_react.py
├─ main
│   └─ ...
```

```
├─ requirements.txt
├─ setup.cfg
├─ setup.py
└─ test
    └─ ...
```

Tanda `...` merupakan placeholder dari isi folder di atasnya.

Instalasi *Software* Pendukung dan *Environment*

Software Pendukung

Untuk menjalankan aplikasi web, sistem operasi harus sudah terpasang **NodeJS**. **NodeJS** yang diperlukan adalah **NodeJS** versi **10.9.0** atau lebih baru. **NodeJS** dikategorikan ke dalam *software* pendukung karena tidak dibutuhkan pada proses utama dari program ini. **NodeJS** dapat diperoleh dengan menjalankan *binary* yang tersedia pada [halaman unduh NodeJS](#).

Environment Python

Pada folder utama dari program telah tersedia berkas bernama `requirements.txt`. Berkas ini mengandung daftar *software* pendukung yang diperlukan untuk menjalankan program. *Software* pendukung tersebut dibuat dalam bahasa **Python** yang dapat dipasang langsung.

1. **Package Manager** khusus **Python** bernama **pip** harus sudah terpasang pada sistem operasi. **pip** dapat diperoleh dengan mengikuti proses instalasi pada [halaman ini](#).
2. Pastikan **terminal** atau **command prompt** pada sistem operasi telah menunjuk ke folder utama program yang mengandung berkas `requirements.txt`.
3. Jalankan perintah `pip install -r requirements.txt` dan tunggu hingga proses selesai.
4. *Software* pendukung dan *environment* sudah siap bila perintah sukses dijalankan tanpa ada error.

PENGUNAAN PROGRAM

Bagian ini menunjukkan cara menjalankan *interpreter* dan hasil berupa aplikasi web.

Pengunaan *IFML to React Interpreter*

Untuk menjalankan program, berkas IFML berekstensi `.core` dan UML berekstensi `.uml` dibutuhkan oleh *interpreter*. Berkas tersebut harus diletakan pada *folder* yang sama dengan program. Program tersebut berupa berkas `ifml_to_react.py`. Langkah untuk menjalankan *interpreter* adalah:

1. Pastikan **terminal** atau **command prompt** pada sistem operasi telah menunjuk ke *folder* utama program yang mengandung berkas `ifml_to_react.py`, `<nama aplikasi>.core`, dan `<nama aplikasi>.uml`.

2. Jalankan perintah `python ifml_to_react.py` dan tunggu hingga proses selesai hingga menunjukkan pesan `INFO React PWA Project successfully generated at result`.
3. Folder **result/**`<nama aplikasi>` akan terbuat secara otomatis.
4. Hasil dari *interpreter* terdapat dalam folder `<nama aplikasi>`.

```
|— ...  
|— result  
|   |— <nama aplikasi>  
|— ...
```

Konfigurasi

Ketika pertama kali memasang program, program telah siap untuk dijalankan dengan konfigurasi dasar. Konfigurasi tersebut mempengaruhi hasil dari aplikasi web yang akan digunakan.

Nama Aplikasi

`<nama aplikasi>` pada tahap sebelumnya merupakan *placeholder* untuk nama berkas yang diperlukan. Ketika pertama kali memasang program, berkas awal `<nama aplikasi>` yang tersedia adalah `abs_bankaccount`. Pada folder utama dapat ditemukan berkas `abs_bankaccount.core` dan `abs_bankaccount.uml`.

Untuk mengganti nama aplikasi, `abs_bankaccount` diganti menjadi nama yang diinginkan. Hal yang perlu diganti adalah:

- Berkas `abs_bankaccount.core`

- Berkas `abs_bankaccount.uml`
- Sesuaikan variabel `ifml_file` dan `uml_file` pada *line* 15 dan 17 dalam berkas `ifml_to_react.py` dengan nama berkas yang telah diganti.

Google Client ID

Ini digunakan oleh aplikasi web untuk proses autentikasi. Proses autentikasi ini membutuhkan internet dan juga *server* API **ABS Microservice** yang tersedia atau yang dapat dipasang secara local dengan mengakses [affandhia/abs-backend-docker](#). **Google Client ID** harus sama baik pada API *server* maupun pada *interpreter*. Untuk mengubahnya dapat dengan mengganti nilai variabel `client_id` pada *line* 13 dalam berkas `ifml_to_react.py`.

Pengunaan Aplikasi Web dari Hasil *Interpreter*

Hasil dari *interpreter* berupa aplikasi web berbasis React 16 yang didukung oleh NodeJS. Prproses untuk menjalankan aplikasi web dapat mengikuti tahap berikut:

1. Pastikan **NodeJS** versi **10.9.0** atau lebih baru dan *browser* yang mendukung **JavaScript** versi **ES5** atau lebih baru (contoh **Google Chrome** atau **Mozilla Firefox** terbaru).
2. Pastikan API **ABS Microservice** ([affandhia/abs-backend-docker](#)) telah berjalan pada `localhost:7776`.
3. Pastikan **terminal** atau **command prompt** pada sistem operasi

telah menunjuk ke folder **result/** **<nama aplikasi>** .

4. Jalankan perintah `npm install && npm run start:production:clean` .
5. *Browser* standar akan terbuka secara otomatis dan mengarahkan ke halaman <http://localhost:3001/>.

Konfigurasi

Ketika pertama kali memasang program, program telah siap untuk dijalankan dengan konfigurasi dasar. Konfigurasi tersebut mempengaruhi jalannya hasil aplikasi web.

Port yang Digunakan Aplikasi Web

Port standar yang digunakan adalah 3001. Untuk mengubahnya dapat dengan mengganti *port* angka 3001 pada *line* 20 dalam berkas `result/<nama aplikasi>/package.json` menjadi *port* lain yang tersedia.

API ABS Microservice

Program akan secara standar mengatur aplikasi web untuk meminta data kepada *server* API **ABS Microservice** yang tersedia pada `localhost:7776` . Untuk mengubahnya dapat dengan mengganti `http://localhost:7776` pada *line* 20 dalam berkas `result/<nama aplikasi>/package.json` menjadi alamat lain yang menyediakan API.